Biomedical Image Processing – Noise Filtering & Rotational Landmarks

Paul Vu
301169550
March 24th 2017

SFU

Project Overview

In this project, I first filter an image containing unknown periodic noise as well as salt and pepper noise. I do this by evaluating the image in the frequency domain and designing a high pass filter to filter out the periodic noise.

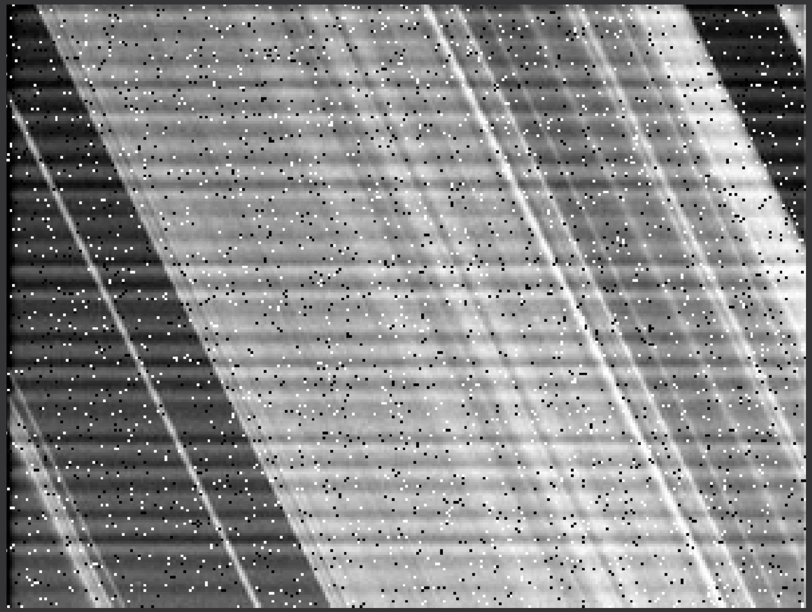

Fig. 1 Noisy Saturn Image to be filtered

Filtering Salt & Pepper

Given the noisy Saturn image I first filter the salt & pepper noise using a 3x3 median averaging filter. I take the surround pixels and average them, which will get simulate a filtering of the salt and pepper. Since I am taking the average , we will still see some residue of salt and pepper since these image values will impact the pixel value by $1/9^{th}$. Although the results aren't perfect, they do filter out almost all of the salt and pepper.

Filtering Periodic Noise

After filtering the Salt & Pepper noise, I then take the image and transform it to the frequency domain. From there I visually evaluate the spectrum and noticed that there were three unusual high frequency lines 1 horizontally, 1 vertically and 1 diagonally. Through trial and error, I found that the periodic noise was caused by the vertical strip of frequency values and then developed a high pass filter that ran down N/2, where N is the column length of the image. The figure below illustrates the results of the filter applied to the image.

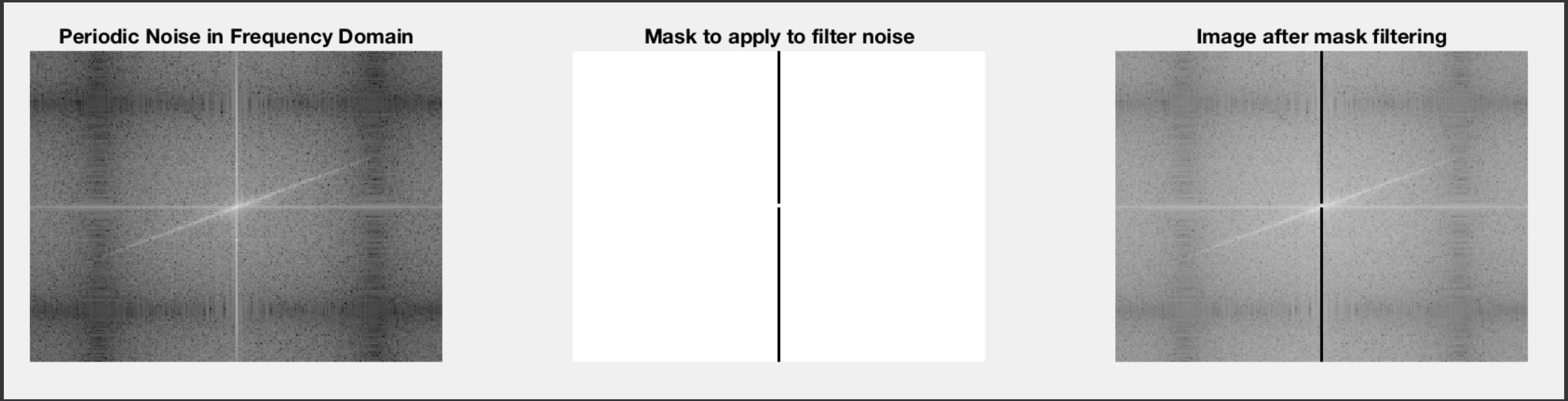

Fig.2 Overview of high pass filter

Saturn Ring Filtering Results

The below figure shows the results of the image after each filter was applied to it. In Figure 3, I used my own Median Filter, which isn't as effective as MATLAB's built in median filter as shown. This is because my mask is only a 3x3 matrix, and is still impacted very slightly by the salt and pepper noise.

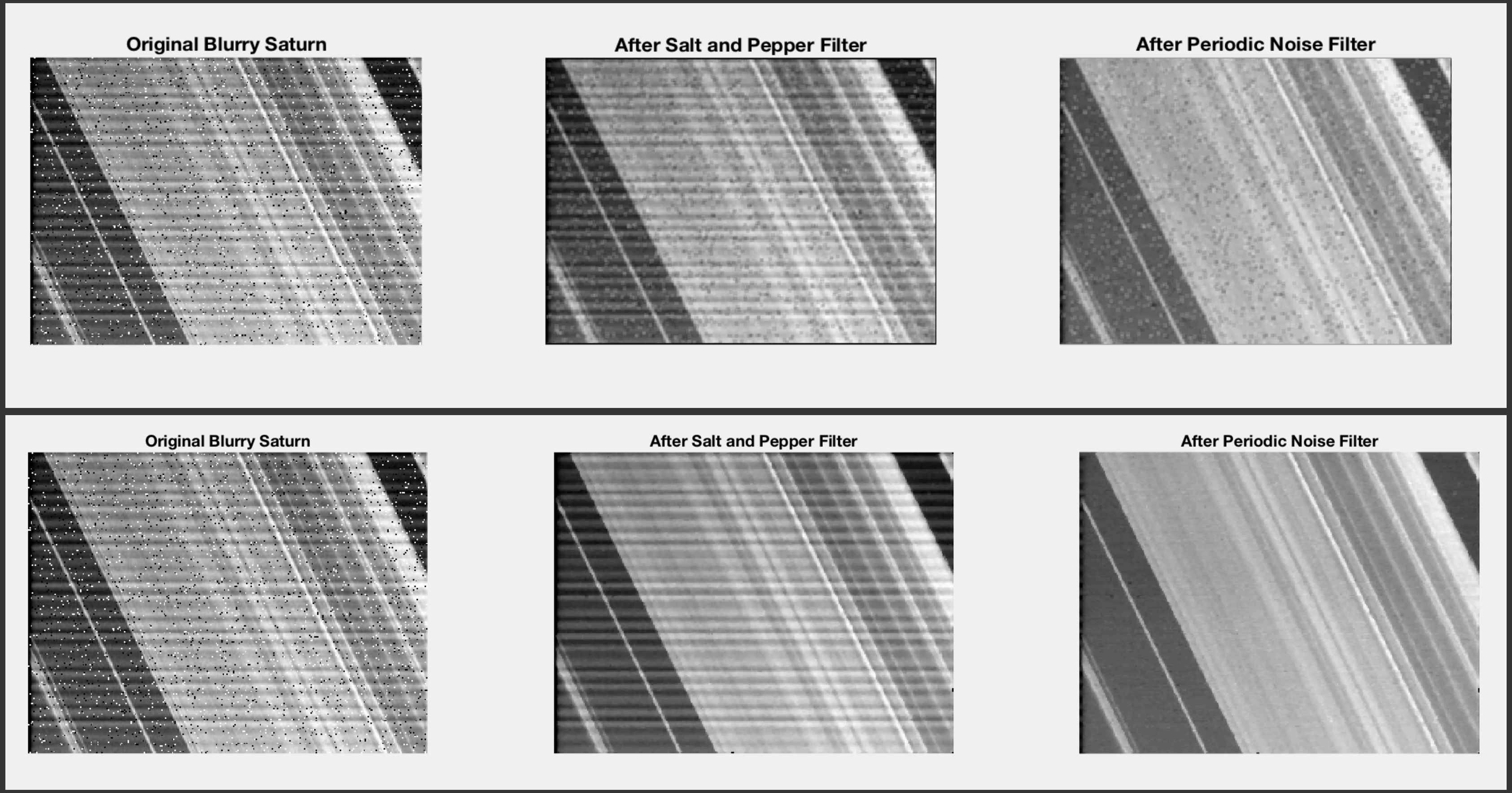

Fig. 3 Results of each filter using custom median Filter (top) and MATLAB's median Filter (bottom)

Image Registration Goal:

Given a two images, Image1 with a straight faced selfie, and Image2 with rotation and different facial expression we would like to investigate how to translate Image2 back to the same orientation of Image1 using image registration. We would like to see how accurate our mapping function is and will evaluate the results by showing the difference between original Image1 and transformation of Image2

Image Registration: Data Set

To analyze the impact or angle of rotation on Image registration I take 3 variations of my selfie photo:

A) Zero rotation from original image
B) Small rotation from original image
C) Large rotation from original image

Fig. 4 From Left to right, Original Image, A, B, C

Image Registration: Process

Using MATLAB's built in "CPSELECT" function, I map each rotation image to the original image collecting a set of landmarks for each image. Results are shown in Fig. 5.

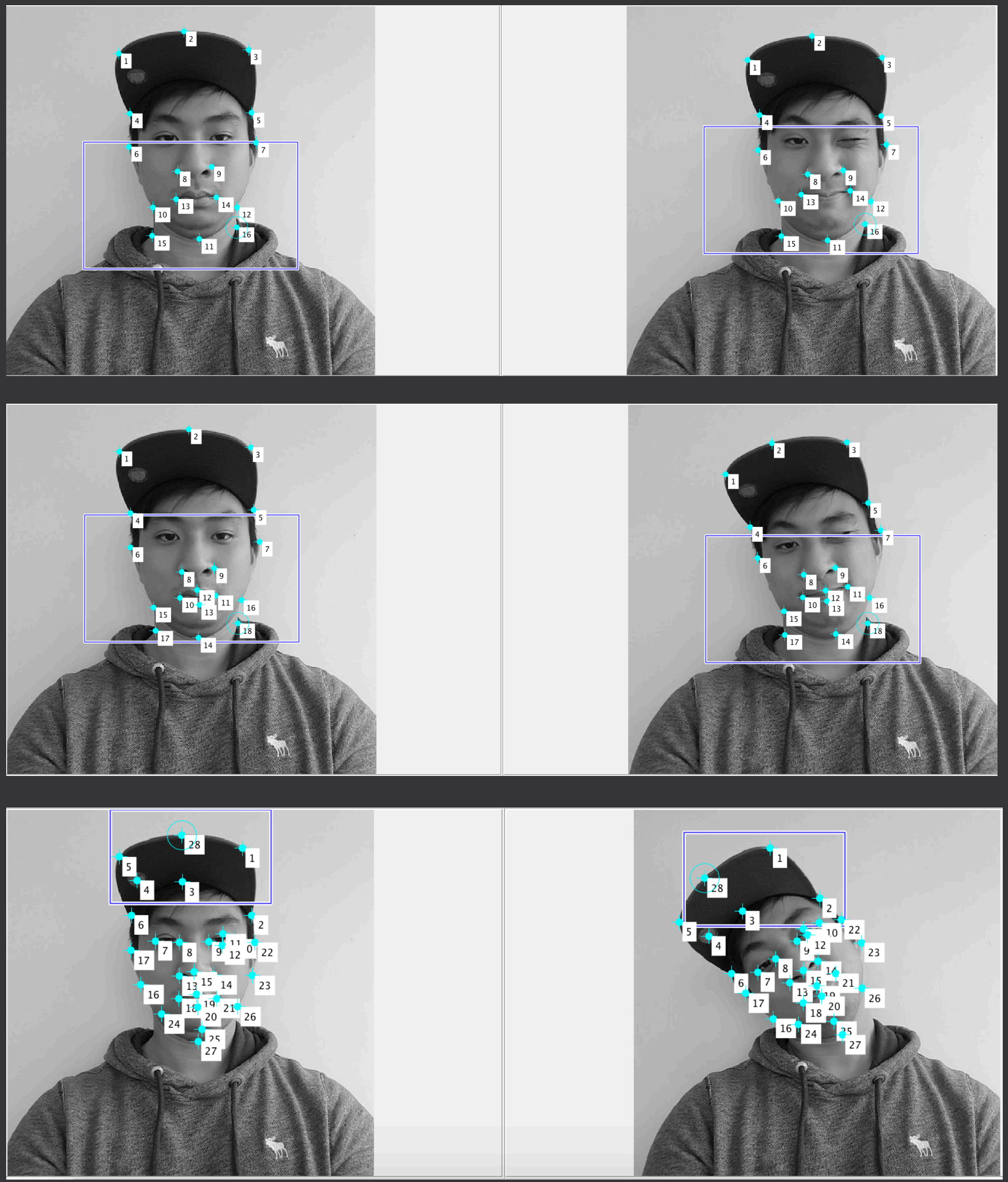

Fig. 5 From top to bottom, Mapping of original Image (left) to A, B, C (right)

Image Registration: Theory & Calculations

After collecting each landmark for the three rotation scenarios, I perform a mapping function to calculate phi such that phi will map each point on the image to a transformed image as depicted in the equations below. The Phi equation will hold the x and y coordinates of the pixel value for the transformed image, which must be interpolated against the rotated image since these will be floating point numbers.

$$\phi(x) = \hat{R}x + \hat{T}$$

where $\hat{R} = USV^t$

$$\hat{T} = \mu_q - R\mu_p$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

p_i = Image1 landmarks

q_i = Image2 landmaks

$$\mu_q = \begin{bmatrix} \text{average x-coordinate q-landmark} \\ \text{average y-coordinate q-landmark} \end{bmatrix}$$

$$\mu_p = \begin{bmatrix} \text{average x-coordinate p-landmark} \\ \text{average y-coordinate p-landmark} \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Equation 1. Registration Transformation Equation

Image Registration Design Choice:

A major design choice in registration is deciding how much of the image to rotate. In other words, how much of the rotated face do we want to crop and rotate to the original orientation? This design choice heavily impacts the quality of registration. In the algorithm implemented here, we use a rectangular cropping method where we crop from the top of the image to the maximum y-value of the landmark.

Our vertical & horizontal constraints are specified by the min and max values in the x-y direction specified by the landmarks of the original image. We add a threshold of 20 to each side for safe measure that we can crop the entire width of the face. This is needed because when a face is rotated, the horizontal length increases in comparison to the original image, since the face is now a diagonal.

The best cropping method would be one which would crop only the rotated part of the face, since using a rectangular method will still crop out the background and shoulders of the person slightly. We see that the larger the angle of rotation, the moor poorly our rectangular cropping method works. Therefore for large values of rotations we would want a more ideal cropping method to improve registration quality.

Image Registration Results: Zero Rotation

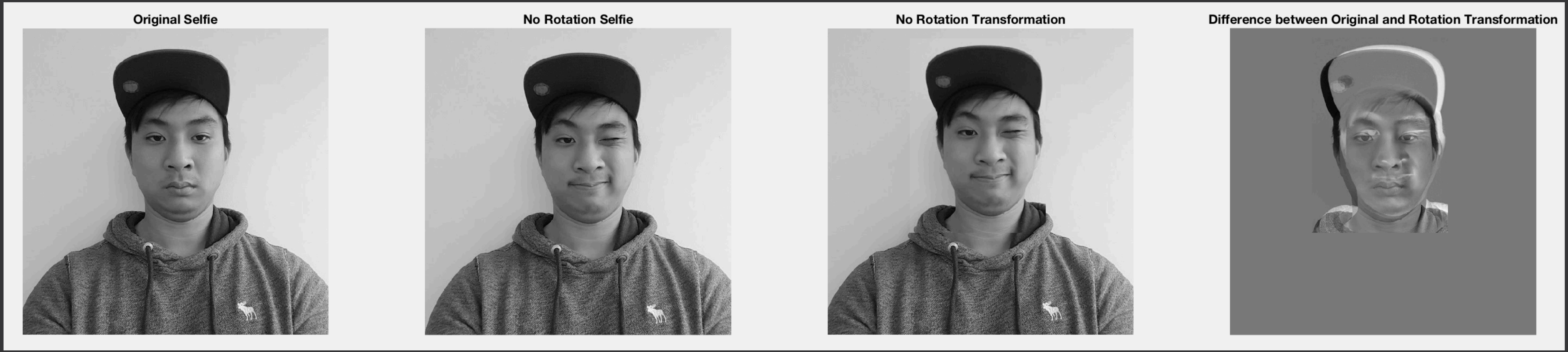

Fig. 6 Image Registration Transformation and Comparison of Zero Rotation

Image Registration Results: Small Rotation

Fig. 7 Image Registration Transformation and Comparison of Small Rotation

Image Registration Results: Large Rotation

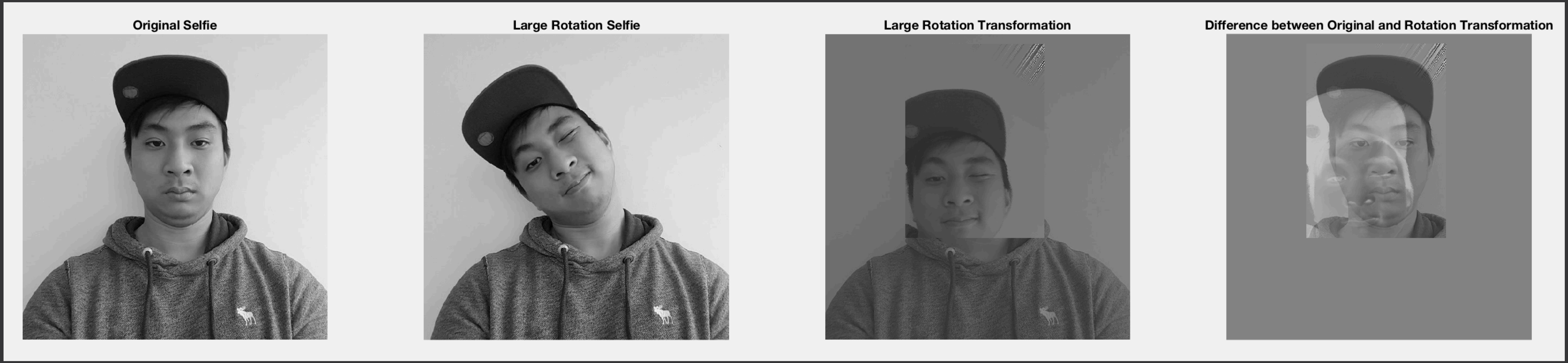

Fig. 8 Image Registration Transformation and Comparison of Large Rotation

Image Registration Analysis

The effect of angle of rotation on image registration is clear. The stronger the degree of rotation the more inaccurate our transformation function is. This is because as our function only tries to map the portion of the image which was rotated, it does so using a “rectangular” box method. Therefore we see sharp cuts off of the neck area from the body which explains why the large rotation is so disjointed compared to the slight rotation and zero rotation.

In the case where there was zero rotation, our results were the best since there was minimal transformation required. Our image was transformed slightly because of my winking and smiling which effected the landmarks displacement from the two images, thus impacting the transformation function. This can be seen in the difference image as my face is displaced slightly to the right as shown by the dark pixels on the left side of my face.

For the slight rotation image, we again see that the transformation is relatively accurate as my face has been rotated to the upright position.

Finally, when we subtract the transformation from the original image, the best case is the image with least amount of rotation since the majority of the image is similar we would see a very dark image (near 0 value) indicating the images are similar. The only values we should see clearly are the differences (smile and winks) from image 1 and 2, however due to over rotation there is some error since my faces don’t line up.

Image Registration Conclusion

To conclude the quality of the registration is dependent on how heavily the rotation is, and the threshold function used to crop the face. If there was a way to only crop the face without using a rectangular method, then the disjointed edges shown in figure 8 will be minimized and our registration quality will be improved.